ESD-TR-86-018

Technical Report
747

Hopfield Model Applied to Vowel and Consonant Discrimination

B. Gold

3 June 1986

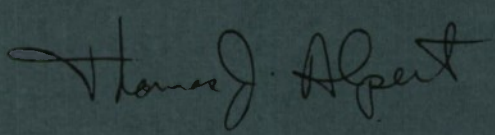
Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Department of the Air Force
under Electronic Systems Division Contract F19628-85-C-0002.

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the Department of the Air Force under Contract F19628-85-C-0002.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER



Thomas J. Alpert, Major, USAF
Chief, ESD Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

HOPFIELD MODEL APPLIED TO VOWEL AND CONSONANT DISCRIMINATION

B. GOLD
Group 24

TECHNICAL REPORT 747

3 JUNE 1986

LEXINGTON MASSACHUSETTS

ABSTRACT

A Hopfield model of 120 "neurons" has been simulated on an LDSP (Lincoln Digital Signal Processor). The model has been applied to the study of problems in automatic discrimination of vowels and consonants. In the first problem, a spectral cross section was extracted by performing a fast Fourier transform on a 20-ms segment from the steady-state portion of the vowel in a single-syllable word. The spectrum was then smoothed and a one-bit gradient measure applied at 120 frequency values, thus obtaining an assigned state for that vowel. This procedure was repeated until eight such assigned states were obtained. From these data, the connection matrix T_{ij} was obtained using the associative equation,

$$T_{ij} = \sum_{s=0}^{7} x_i^s x_j^s$$

Each x_i^s was the component of one of the assigned states.

A discrimination experiment was performed by specifying, as input, half of the 120 bits for each state. This specification was performed randomly, 8 times for each of the 8 states, resulting in a set of 64 runs. For 51 of these runs, the convergence to the correct state was perfect. Of the 13 spurious states generated, most were closest to the correct assigned states.

Using the same T_{ij} matrix, a second experiment was performed. Each of the words containing a given vowel was scanned and a new input pattern was obtained every 10 ms. Each of these patterns was applied as an initial state to the neural network, which was allowed to iterate until convergence. Many (but not all) of the vowel cross sections were most closely identified by the network as the assigned state associated with that word. Outside of the vowel, results were random.

The final experiment was an initial attempt to automate the Diagnostic Rhyme Test that is widely used to rate speech-processing devices such as vocoders. In this test, the listener must choose between two initial consonants (e.g., pool vs tool) that generally differ in only a single distinctive feature. Several algorithmic additions to the Hopfield model have been tried; to date, the most successful one is a two-state model that has some fore-knowledge of the critical intervals of the rhyme pair. Work is continuing on this problem.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

HOPFIELD MODEL APPLIED TO VOWEL AND CONSONANT DISCRIMINATION

1. INTRODUCTION

Although the study of both real and artificial neural networks has been a subject of continuing interest for many years,[1,2] recent work by Hopfield and associates has catalyzed a new spurt of activity.[3-5] In particular, Hopfield's proposed model of associative memory has raised questions concerning the applicability of the model to traditional problems such as speech and image recognition. This report describes some computer simulation results on vowel and consonant discrimination based on this associative memory model.

Section 2 of this report reviews several properties of the associative memory model. Section 3 presents the results of a Hopfield network simulation that performs vowel discrimination among eight vowels when random errors perturb the representation. Section 4 describes further work where the network scans an entire word, searching for vowels that are similar to one of the stored vowels. Section 5 describes an approach to the consonant discrimination problem and some preliminary results. Section 6 is a brief description of the simulation facilities.

2. SOME PROPERTIES OF HOPFIELD NETWORK ASSOCIATIVE MEMORIES

Figure 1 represents a digital simulation of an N^{th} order Hopfield network. Each element is described by a first-order difference equation. This results in a set of graded outputs $x_0, x_1 \ldots x_i \ldots x_{N-1}$. Inputs to any one element come from all the other N-1 elements via a **transformation** or **connection** matrix T_{ij} ($T_{ii} = 0$). Each graded output x_i is passed through a memoryless nonlinear element (nℓe) to generate the signals $\hat{x}_i$. Figure 2 shows some candidate nonlinear elements.* Figure 3 shows the operation of a single element in this array of N elements. The **memory** of the system of Figure 1 resides in the matrix T_{ij} via the following procedure:

$$\text{Let } X^0 = (\hat{x}_0^0, \hat{x}_1^0, \ldots \hat{x}_{N-1}^0)$$

$$X^1 = (\hat{x}_0^1, \hat{x}_1^1, \ldots \hat{x}_{N-1}^1)$$

$$\vdots$$

$$X^{M-1} = (\hat{x}_0^{m-1}, \hat{x}_1^{m-1}, \ldots \hat{x}_{N-1}^{m-1})$$

The X^k define a set of **assigned** or **stored** states. The network will behave as an associative memory if

$$T_{ij} = \sum_{s=0}^{m-1} \hat{x}_i^s \hat{x}_j^s \tag{1}$$

Given this "recipe" for the T_{ij} matrix, then, under appropriate conditions, the network will evolve to one of the assigned states even if the initial state is other than a stored state. That is, the state of the network will change with each iteration until its state becomes one of the **stored** states, at which point it will remain in that state. Thus, the **assigned** states are **stable** states of the network.

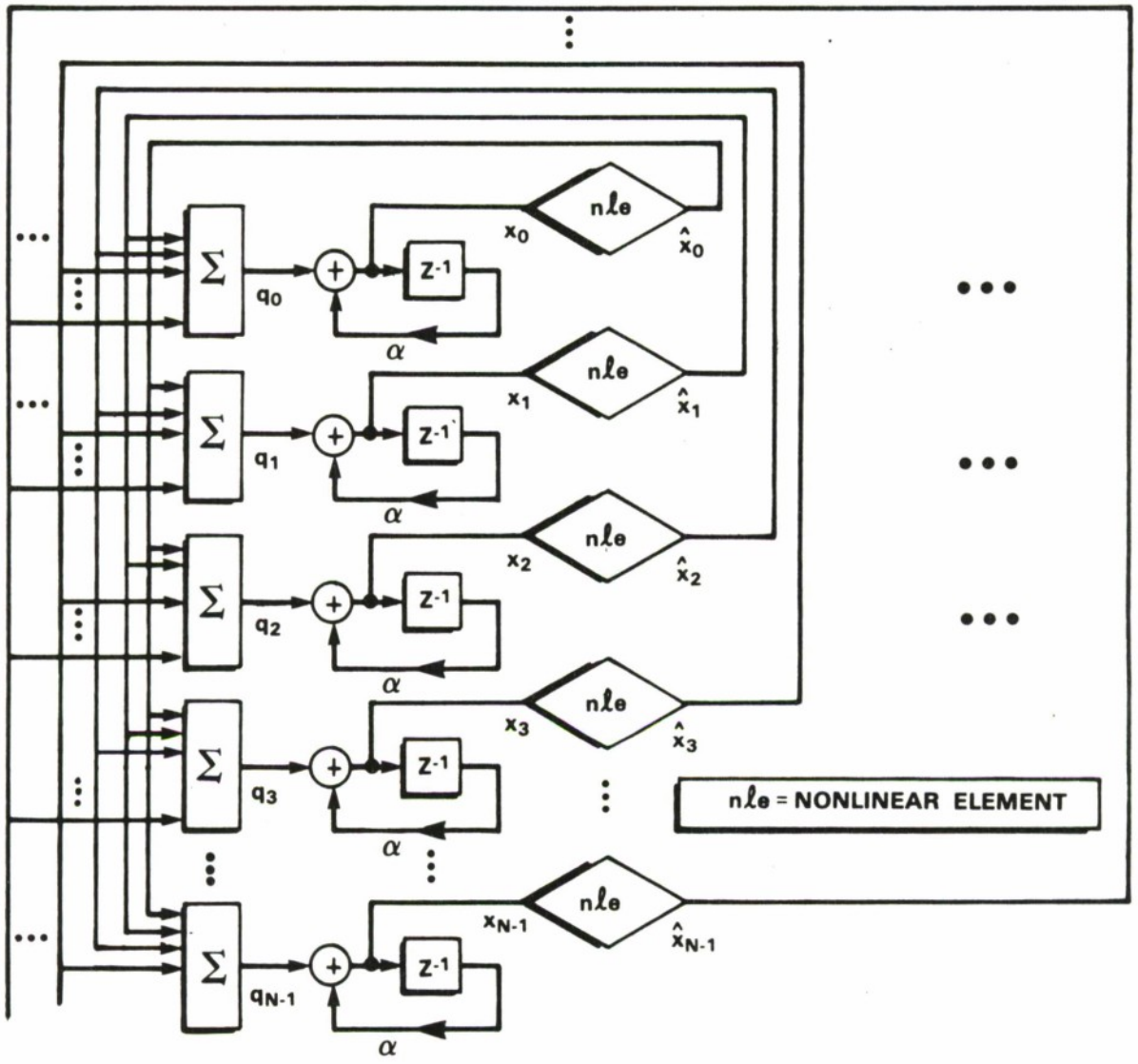
The algorithm that describes the evolution of the network is given by the equations

$$q_i(n) = \sum_{j=0}^{N-1} T_{ij}\, \hat{x}_j(n-1) \tag{2a}$$

$$x_i(n) = \alpha\, x_i(n-1) + q_i(n) \tag{2b}$$

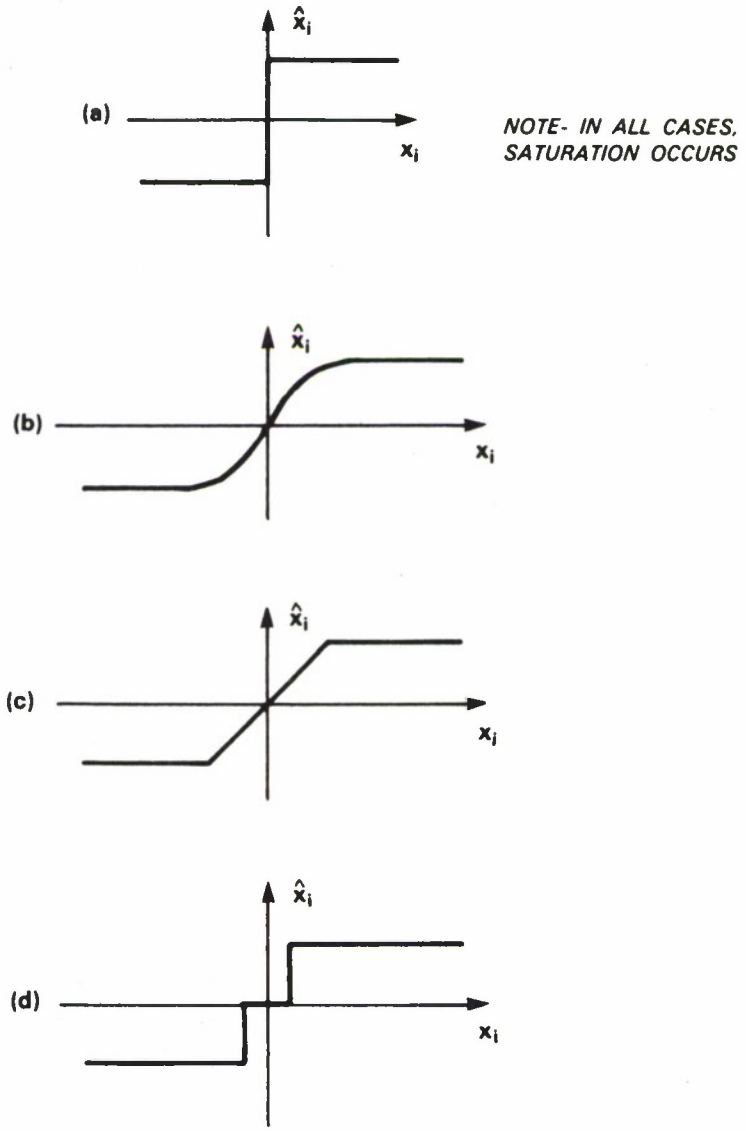
$$\hat{x}_i(n) = n\ell e[x_i(n)] \tag{2c}$$

In these equations, n represents discrete time samples, α is the time constant of the unity gain (at zero frequency) first-order digital filter. (In all our simulations α has been set to 0.75.) nℓe represents the function of the nonlinear element of Figure 2(a) which we use in Figures 1 and 3.



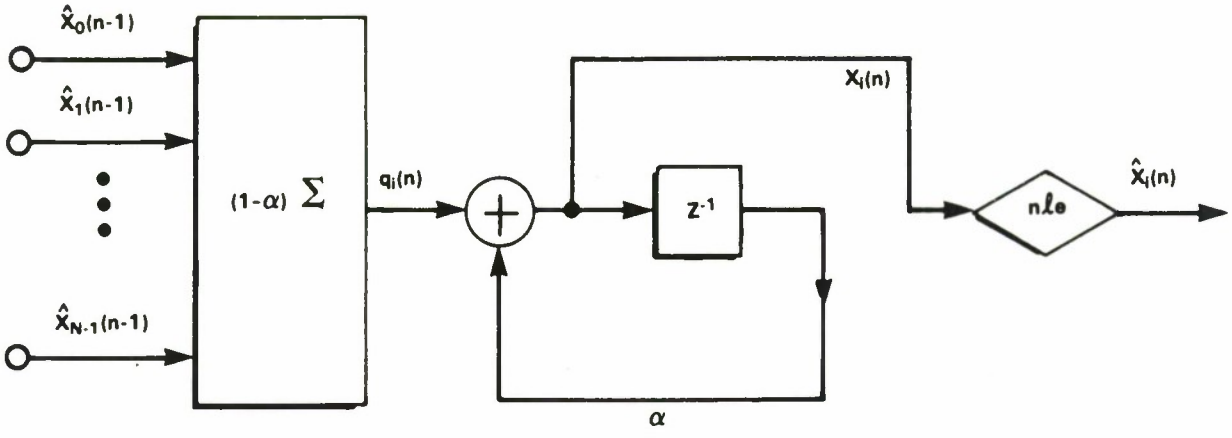
74319-2

Figure 1. Digital simulation of a Hopfield net. Network ***state*** *is defined by the N values* $\hat{x}_0, \hat{x}_1, \ldots \hat{x}_{N-1}$.



74319-3

Figure 2. Some representative nonlinear functions of a "neural" element.



$$q_i(n) = (1-\alpha) \sum_{j=0}^{N-1} T_{ij}\hat{X}_i(n-1),\ T_{ii} = 0$$

$$X_i(n) = \alpha\ X_i(n-1) + q_i(n)$$

74319-4

Figure 3. Operation of a single element in a Hopfield net.

In general, the network can drift to **spurious** states; that is, stable states that are **not** one of the assigned states. The precise relationship between a network and its spurious states is not well understood. A useful rule of thumb is this: for $m < 0.15N$, there is a high probability that the network will converge to an assigned state. For $m > 0.15N$, there is a high probability that the network will converge to a spurious state.

If the initial state of the network is set by some external input that is, in some sense, reasonably close to one of the assigned states, the network has a good chance of converging to that assigned state. Another way to say this is that the network can recognize memorized states that are only partially described. This basic property makes it possible for the network to correctly identify a noisy input and also to continue to operate successfully when individual elements are inoperative.

* See Reference 1 — results in this report are based on the nonlinear element of Figure 2(a).

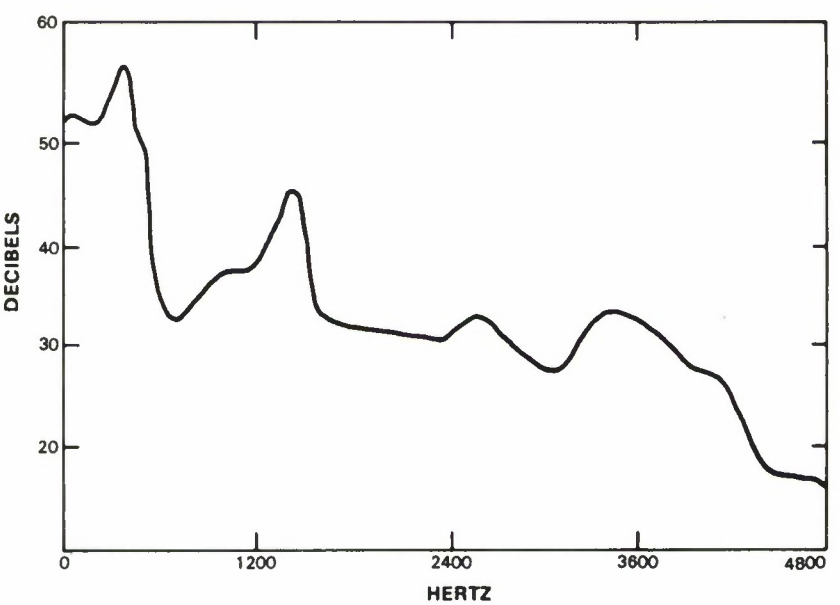
3. APPLICATION TO SIMPLE VOWEL DISCRIMINATION

In order to apply a Hopfield net to the processing of real data, a suitable representation of those data must first be found. In the case of vowel discrimination, such a representation must be derived from the spectrum. There is much evidence that the peripheral auditory system (8th nerve) of humans performs such an analysis before relaying to higher levels. Our simulation has been restricted to Hopfield net elements with binary outputs; that is, in Figure 1, the variables $\hat{x}_i$ are permitted to take on only the values ± 1. Thus, we need to find a suitable **binary** representation of the spectral cross section of a vowel. Our procedure is the following.

a. Perform a high-resolution spectrum analysis on overlapping segments (of 20-ms duration) of a spoken word. This analysis is performed every 10 ms and the signal is first multiplied by a Hamming window before being analyzed. Analysis is accomplished via a 512-point fast Fourier transform, from which a 256-point magnitude function covering the range 0 to 5 kHz can easily be derived.

b. This magnitude function is then smoothed in frequency to produce a **spectral envelope** function such as shown in Figure 4 for a specific cross section of the steady-state portion of the vowel in the word "vill."

c. Finally, a suitable pattern (for entry into the associative memory) is generated in the following way: let S(f) be the magnitude vs frequency function. Let $S(f_i)$, i = 0,1,...119 be samples of this function. Then, if $S(f_i) > S(f_{i-1})$, set the variable x_i in Figure 1 to +1. If $S(f_i) < S(F_{i-1})$, set x_i to −1. If $S(f_i) = S(f_{i-1})$, randomly choose +1 or −1.



74319-5

Figure 4. Smoothed spectral envelope function for vowel in "vill."

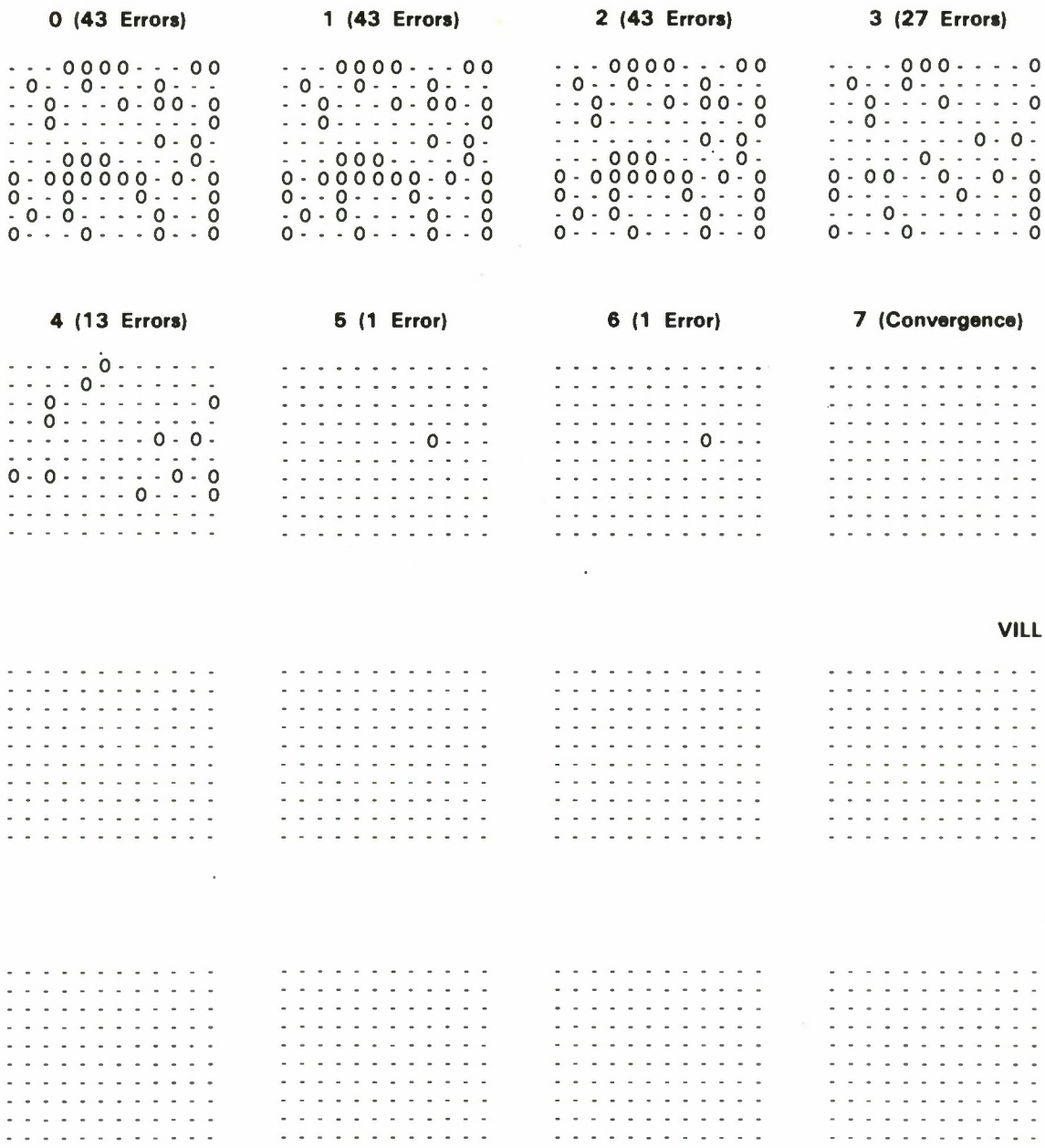


Figure 5. Evolution of a Hopfield net in vowel discrimination.

(It should be noted that a variety of representations can be invented that lead to a binary sequence suitable for input to the Hopfield net. We shall see for instance that a different representation is used for processing of consonants.)

The next task is creation of the network by using Equation (1) to compute the T_{ij} matrix. In our case, the chosen parameters were m = 8 and N = 120. The parameter m defines the number of stored states. Since $m < 0.15N$, our rule of thumb says that convergence to a stored state is to be expected. The waveforms and spectra of eight words (listed as the first of the word pairs in Figures 8 to 12) were examined and for each word, a steady-state region was identified by inspection. Then, a single spectral cross section was subjected to the representation process described above and, via Equation (1), the T_{ij} matrix was computed and became a fixed part of the Hopfield net.

The main purpose of this experiment was to determine how accurately a given vowel must be represented in order to successfully retrieve its prototype from the network. Another way to ask this question: How much **partial** information is needed in order to retrieve the **total** information? (the stored representation of the correct vowel). If we chose as the **input** or **initial** state one of the unadulterated stored states, then we would be supplying **all**, or 100 percent of the required information. If, on the other hand, we supplied 50 percent of the required information and allowed the remainder to be supplied randomly, we would, on the average, be supplying 75 percent of the correct bit pattern. In a formula,

$$PI = 1 - 2\,ER \tag{3}$$

where PI is the resulting partial information based on a given fractional error rate.

In our experiment, an **initial** state was derived from one of the stored states by randomly creating errors in the 120-bit representation of that stored state. The system was then subjected to the iterations defined by Equation (2) until **convergence**; i.e., the state of the system remained constant despite continued iterations. An example of the evolution of the network is shown in Figure 5. For convenient viewing, the 120 bits are presented as a 10×12 matrix. Also, Figure 5 displays the **difference** between any state and the stored target state. Thus, iteration 0 (upper left-hand corner) of Figure 5 displays the difference pattern between the **perturbed** stored state (which is also the initial state of the network) and the unperturbed stored state from which it is derived. Each iteration is numbered and we see that convergence occurs at iteration 7; the network now **stays** in the original unperturbed stored state despite repeated application of Equation (3).

We note that no change at all took place in the first two iterations and that the real "action" occurred at iterations 3, 4, and 5. This result agrees with Hopfield's remark[3] that the network converges in several iterations. In our simulations, implementation of the initial state was accomplished by setting the initial values of the x_i to a large positive or negative value (depending on the sign of the assigned state components $\hat{x}_i$). Due to the time constant of the first-order digital filter, the network was slow in getting started.

Figure 6 shows the evolution for the same initial vowel with a different random error generator. In this case, although the net began with fewer errors than for Figure 5, the system converged to a **spurious state** with Hamming distance 4 from the target state. This demonstrates that the behavior of the network is **not** like that of a correlator or matched filter looking for minimum Hamming distance but a more complicated device. One of the challenges of this work is to understand in finer detail how the system evolves under different conditions.

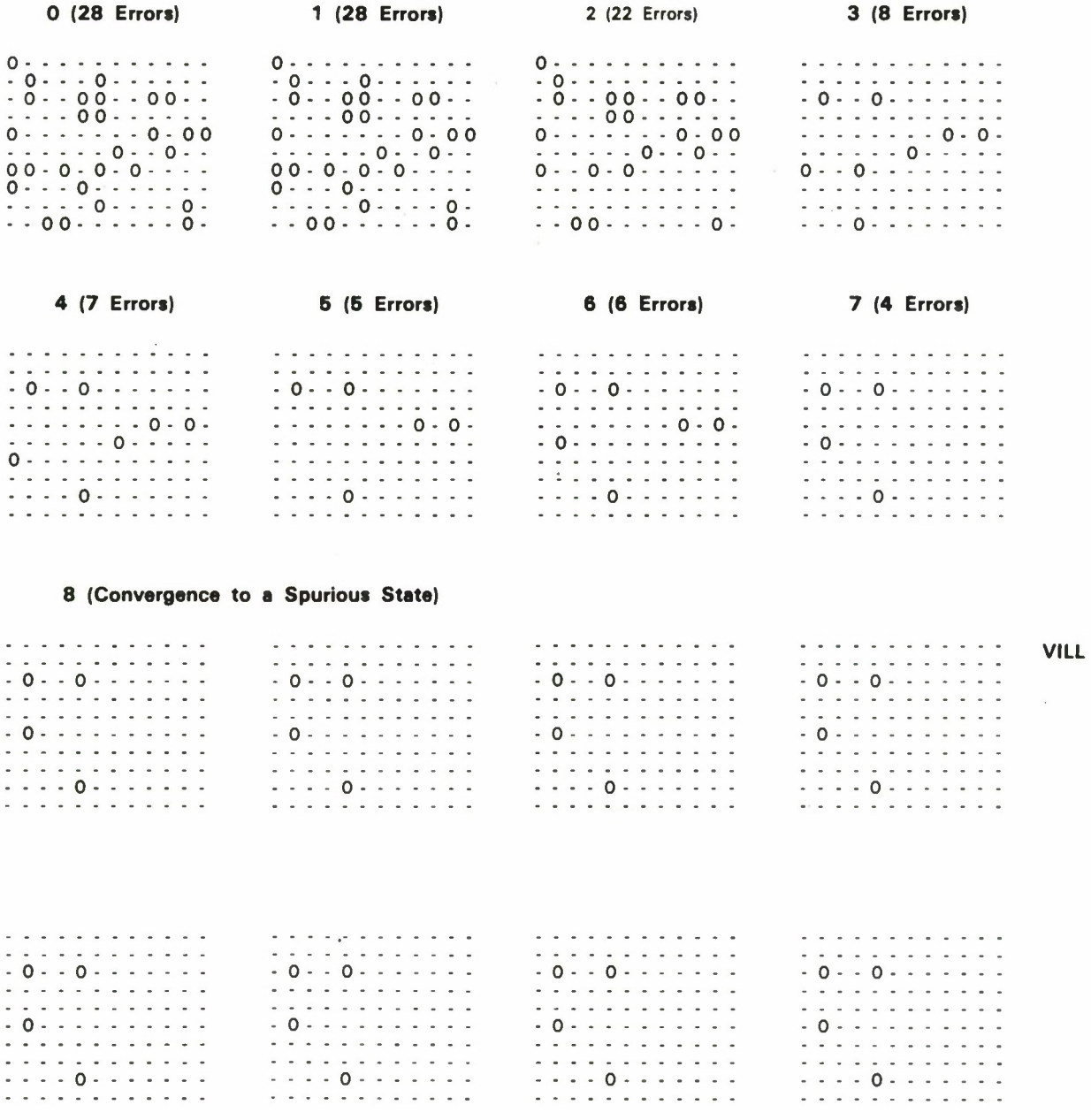
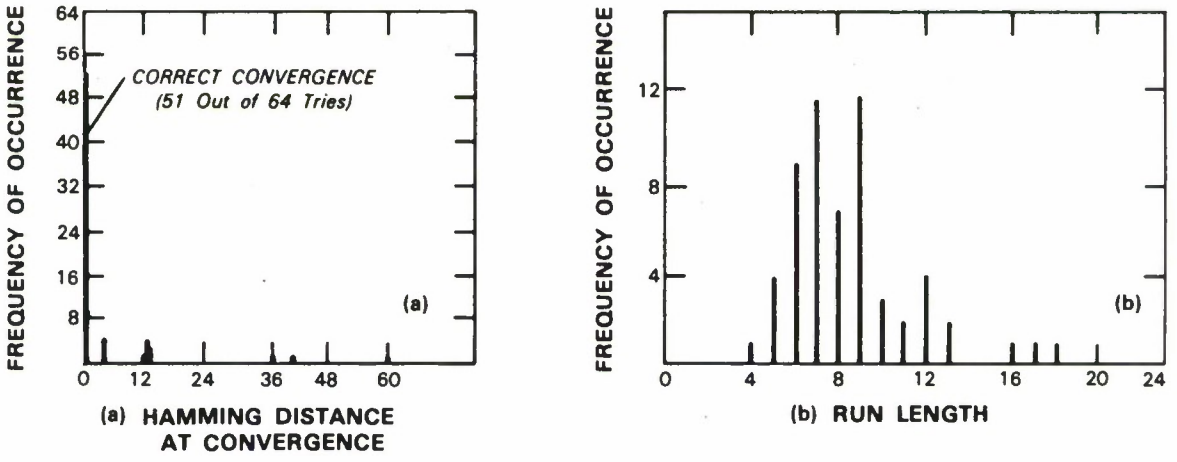


Figure 6. Evolution of a Hopfield net with different initial conditions.

The overall performance of this simple vowel discriminator is shown in Figure 7. Each of the 8 vowels was randomly perturbed with 25 percent error rate. Since different sets of random errors create different results, we ran each vowel 8 times, for a total of 64 runs. Figure 7(a) shows that convergence to the target stored state took place 51 out of 64 times. Usually, when convergence was to a spurious state, that state was **closest** (using a Hamming distance measure) to the target state, but several times the network converged to a different vowel.

Figure 7(b) gives the frequency of occurrence of the run length; that is, the number of iterations to convergence. We see that on the average, about 8 to 10 iterations were needed for convergence.



74319-8

Figure 7. Overall performance of simple vowel discriminator.

4. CONVERGENCE OF NEW VOWELS TO A "FAMILIAR" STATE

Figure 8 shows the waveforms of the two words "vill" and "bill." Each line corresponds to 150 ms so it can be seen that the steady-state portion of the vowels in these words last for several hundred milliseconds. However, it is well known that the measured spectra during vowels do not remain constant; both initial and final consonants can strongly affect this measurement. Furthermore, the vowel in "bill" is modified by a **different** initial consonant than the same vowel in "vill."

On the fourth line of Figure 8 is a small asterisk. This signifies that 20-ms portion of the signal for which an assigned state is defined. For each word pair of the list on the right of Figure 7, a 20-ms portion defines an assigned state. Thus, we construct a network using Equation (1);

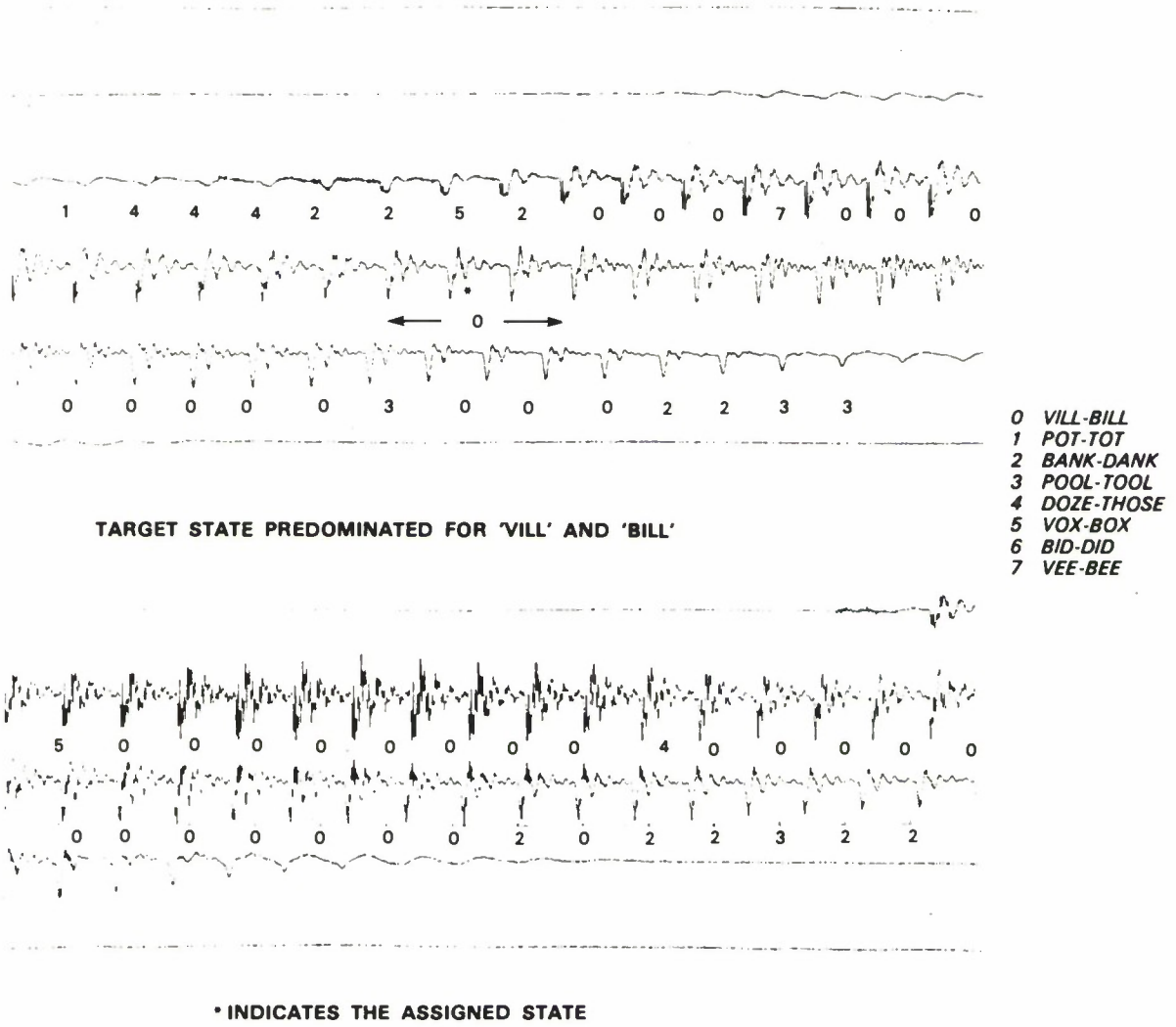
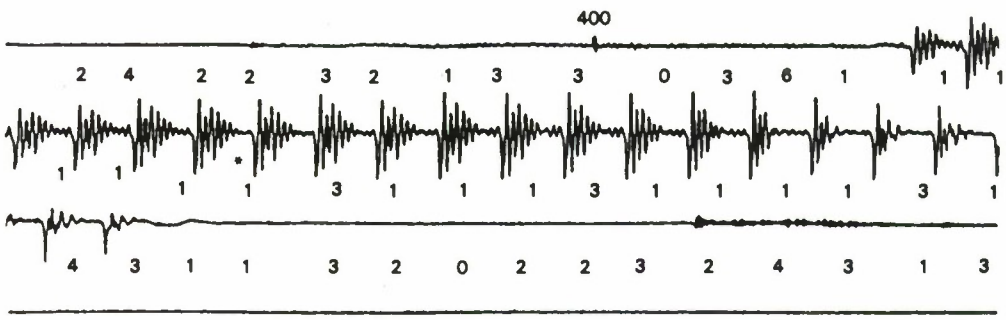


Figure 8. Convergence of new vowels to a "familiar" state.



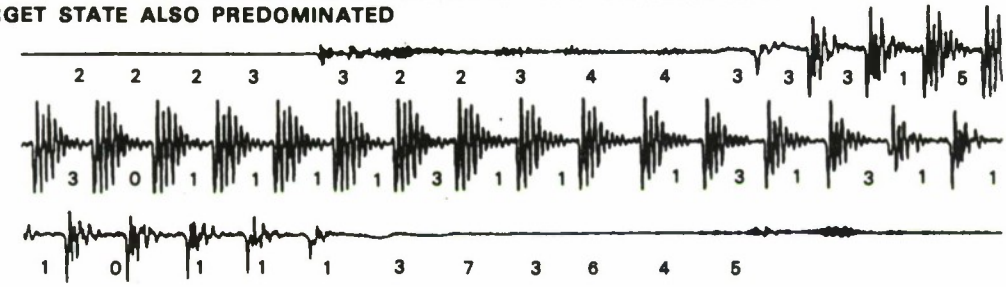
COMMENTS:

1. **'TARGET' STATE IS '1' (Pot)**

2. **DURING STEADY PORTION OF VOWEL, 'TARGET' WAS ACHIEVED ALMOST ALWAYS**

3. **DURING REMAINDER OF THE WORD, RESULTS WERE RANDOM**

4. **DURING STEADY-STATE PORTION OF THE RHYME WORD 'TOT', CONVERGENCE TO THE TARGET STATE ALSO PREDOMINATED**



0 VILL-BILL
1 POT-TOT
2 BANK-DANK
3 POOL-TOOL
4 DOZE-THOSE
5 VOX-BOX
6 BID-DID
7 VEE-BEE

RESULTS FOR 'POT'-'TOT'
***INDICATES THE ASSIGNED STATE**

74319-9

Figure 9. Convergence for the word pair "pot"-"tot."

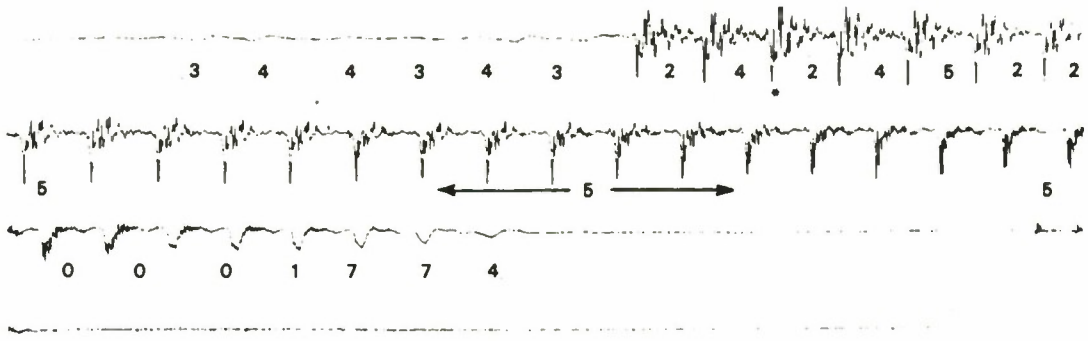
indeed, this is the **same** network we constructed for the experiment of Section 3 but now the task is different. The initial states are now obtained by scanning the entire word pair; every 10 ms a new representation is entered into the network as an initial state and the network iterates until convergence to an assigned or spurious state. At convergence, the final state is compared with all eight assigned states and the assigned state that is minimum Hamming distance is declared to be the "winner." The numbers in Figure 8 show the sequence of winners every 10 ms. We see that in Figure 8 the network has found many "familiar" states, that is, states that were identified as the vowel "i" (as in "vill" or "bill"). This "recognition" was based entirely on the single asterisked segment shown in Figure 8. (It should be pointed out that both "vill" and "bill" and, indeed, all other word pairs processed for this section, were the utterances of a single speaker. Recognition of "familiar" patterns across speakers has not yet been tried.)

Figure 9 shows analysis results for the word pairs "pot" and "tot." Comments are included on the figure. It is seen that results similar to those of Figure 8 were obtained.

Figure 10 shows a case where the network did **not** succeed in recognizing familiar patterns. The target state was 2 (bank-dank), but this was rarely attained. This result can be explained by observing the position of the chosen assigned state (asterisk on line 1). The beginning of the vowel is in the transition period where the initial consonant has the greatest effect, resulting in a spectrum that is grossly different from the steady-state portion of the vowel. **Why** the network converged to (or was closest to) the indicated target states is not clear.

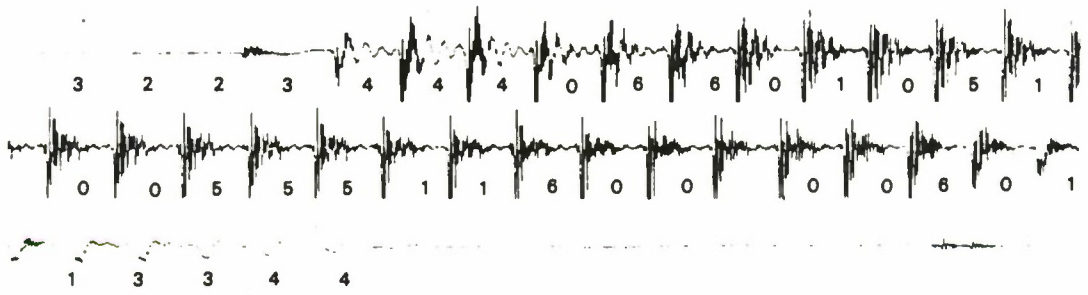
Figure 11 is a repeat of Figure 10 except that the asterisk has been moved closer to the center of the vowel. The results show that many more "familiar" vowels are identified, although in "dank," there are still many "unfamiliar" answers.

The results in Figure 12 are particularly interesting because the vowel (bid-did) is the same as in (vill-bill). The network converged to one or another of these assigned states with comparable frequency. This indicates that the same vowel but from a different context can still be classified as "familiar."



RESULTS WERE WRONG. TARGET STATE 2 WAS RARELY ATTAINED

REASON: ASSIGNED STATE WAS IN THE TRANSITION PERIOD OF THE VOWEL AND SPECTRUM CHANGED GREATLY IN STEADY PORTION

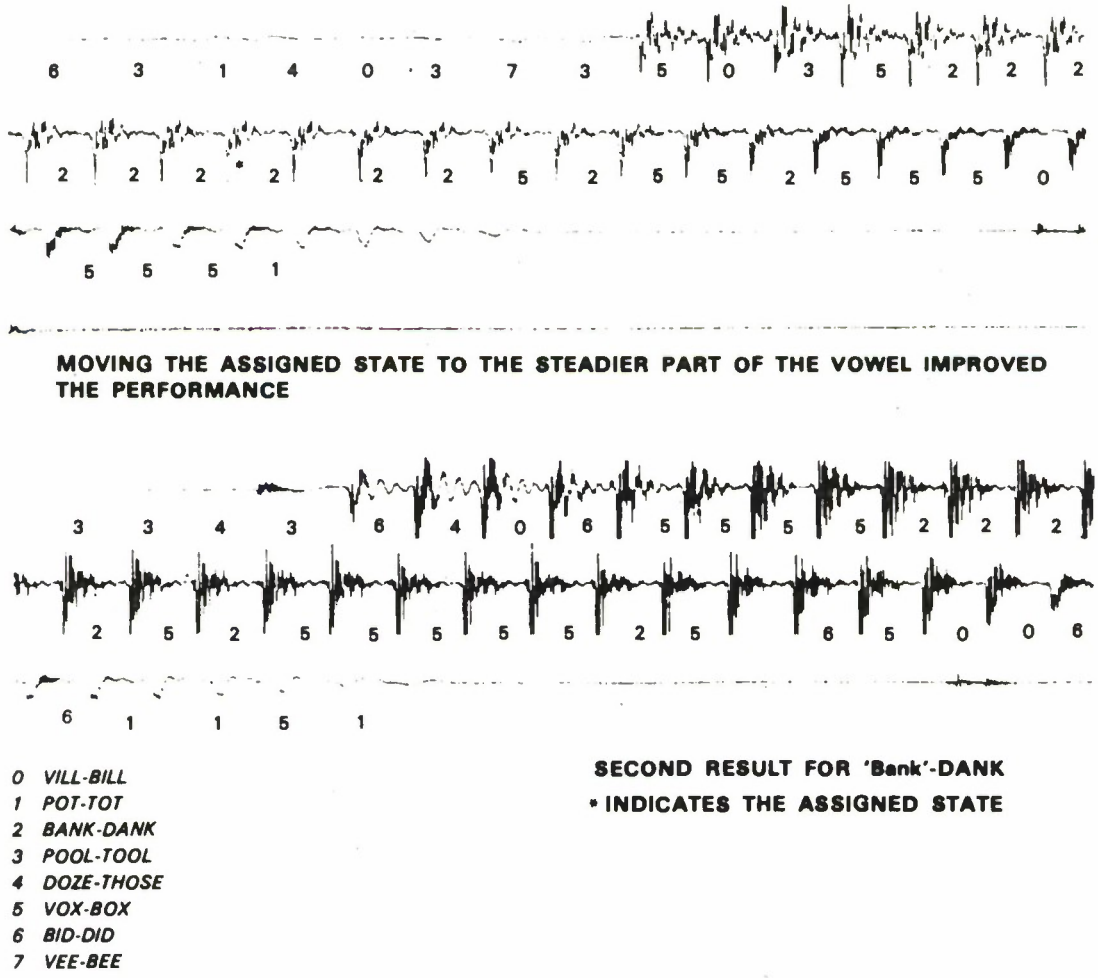


0 VILL-BILL
1 POT-TOT
2 BANK-DANK
3 POOL-TOOL
4 DOZE-THOSE
5 VOX-BOX
6 BID-DID
7 VEE-BEE

RESULTS FOR 'BANK'-DANK
• INDICATES THE ASSIGNED STATE

74319-12

Figure 10. Example where the network failed to recognize familiar patterns.



74319-13

Figure 11. Repeat of Figure 10 with stored states moved closer to the steady-state portion of the vowel.

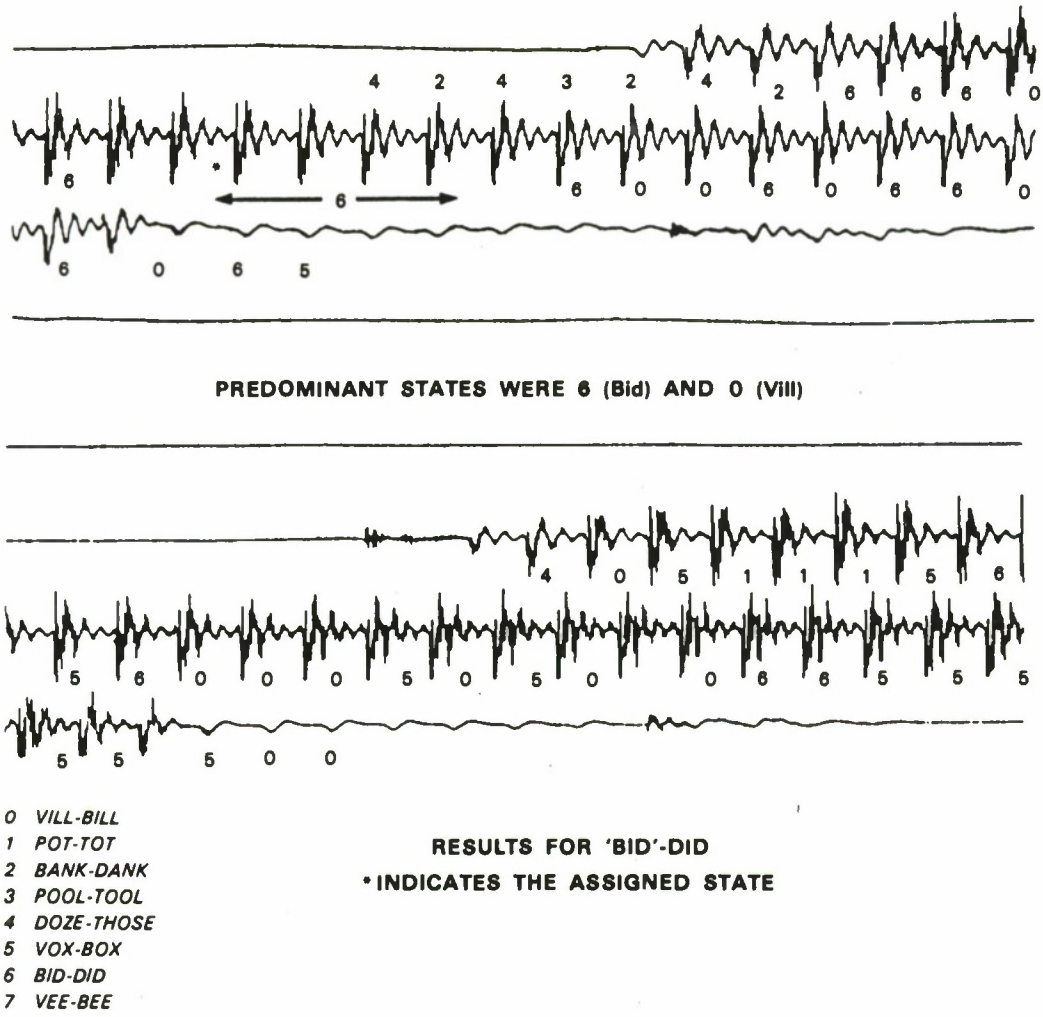


Figure 12. Example of familiarity recognition for the pair "bid"-"did."

5. CONSONANT DISCRIMINATION WITH A HOPFIELD NET

We have seen that vowels tend to have "steady-state" regions during which there is little change in the vowel spectrum. Most consonants do not obey this model but usually involve rapid changes in the spectrum, especially during the transition from the consonant to the following vowel. Also, consonants almost always have appreciably less energy than vowels so that noise and other disturbances can more easily mask the intelligibility of such sounds. For these reasons, it is expected that classification of consonants by Hopfield nets (or any other method) is appreciably more difficult than classification of vowels. For example, it did not seem reasonable to use the same algorithms or parameters for consonants as for vowels. Instead, it seemed more appropriate to study the ability of a Hopfield net to perform the Diagnostic Rhyme Test (DRT).

DRT is one of the classical methods used to assess the intelligibility of speech communication systems. The listener is presented with a list of about 200 rhymed word pairs (e.g., pool-tool, bank-dank) and is asked to choose which of the pairs she/he heard. **DRT** score is defined as,

$$\text{DRT Score} = \frac{\text{correct decisions} - \text{wrong decisions}}{\text{total decisions}} \times 100$$

Thus, a completely random choice would result in a score near zero and a "perfect" score would be 100. There are much data available on the DRT. Typically, high-fidelity speech in a quiet acoustic background would yield a score of 98, whereas a 2400 bps vocoder subjected to the DRT words in a noisy environment (0 to 6 dB signal-to-noise ratio) might yield a score of 60 to 70.

We asked: Could a structure based on Hopfield nets perform an automatic DRT that would compare favorably with the documented results using human listeners? Would this structure bear any resemblance to structures found in simple animals that make yes-no decisions?

We began our work with the (plausible) assumption that there is a relatively small time segment in each of the word-pairs during which a distinction can be made. This segment is either embedded in the consonant or in the transition region between consonant and vowel. At the outset we felt it expedient to make a very significant compromise. Whereas a human listener can focus on the critical intervals "on the fly," we found it necessary to give our system an *a priori* determination of these critical regions. That is, for this experiment we have bypassed the very important problem of "attention," deferring it until we have better understood the problem of binary decision between two consonants that generally differ in only a single distinctive feature.

Our procedure begins with a representation based on a 512-point FFT followed by smoothing in the frequency domain to produce a spectral envelope. Thus far, this is the same as the vowel representation but now, rather than find 120 binary values based on the spectral gradient, only 30 such values are found. The 30 samples are spread over the same frequency range as before, namely 100 Hz to about 5 kHz. However, four **contiguous** such measurements are made in four successive spectral frames, spaced by 10 ms, and the four results of 30 bits each are concatenated to generate a pattern of 120 bits. Thus, we have the same value of N as before but it is based on four consecutive spectral cross sections, or about 50 ms of speech.

The next step is to define the appropriate decision network. But there is no need for us to employ a single network to make decisions for all word pairs. In fact, it seems reasonable to use a separate **two-state** network for each distinct DRT pair. This results in a value of m = 2 for N = 120, which seems like a case of overkill (given the rule of thumb that $m \leqslant 0.15N$). However, given the conditions of most practical DRTs where the speech is processed through a low-rate vocoder or a noisy environment, these numbers did not seem implausible. Furthermore, in some sense it emulated the conditions of a human listener taking a DRT, who also simply needs to make a binary choice for each word she/he hears.

Our data base was the set of word pairs listed in Table I, spoken by a single speaker. For each of the 32 words, a complete pattern of 120 bits was generated every 10 ms and these results were printed, as well as the average spectrum plus some measure of the spectral **change** from frame to frame. These data, plus observation of the waveforms, allowed for a good guess as to

TABLE I
Word Pairs
by Single Speaker

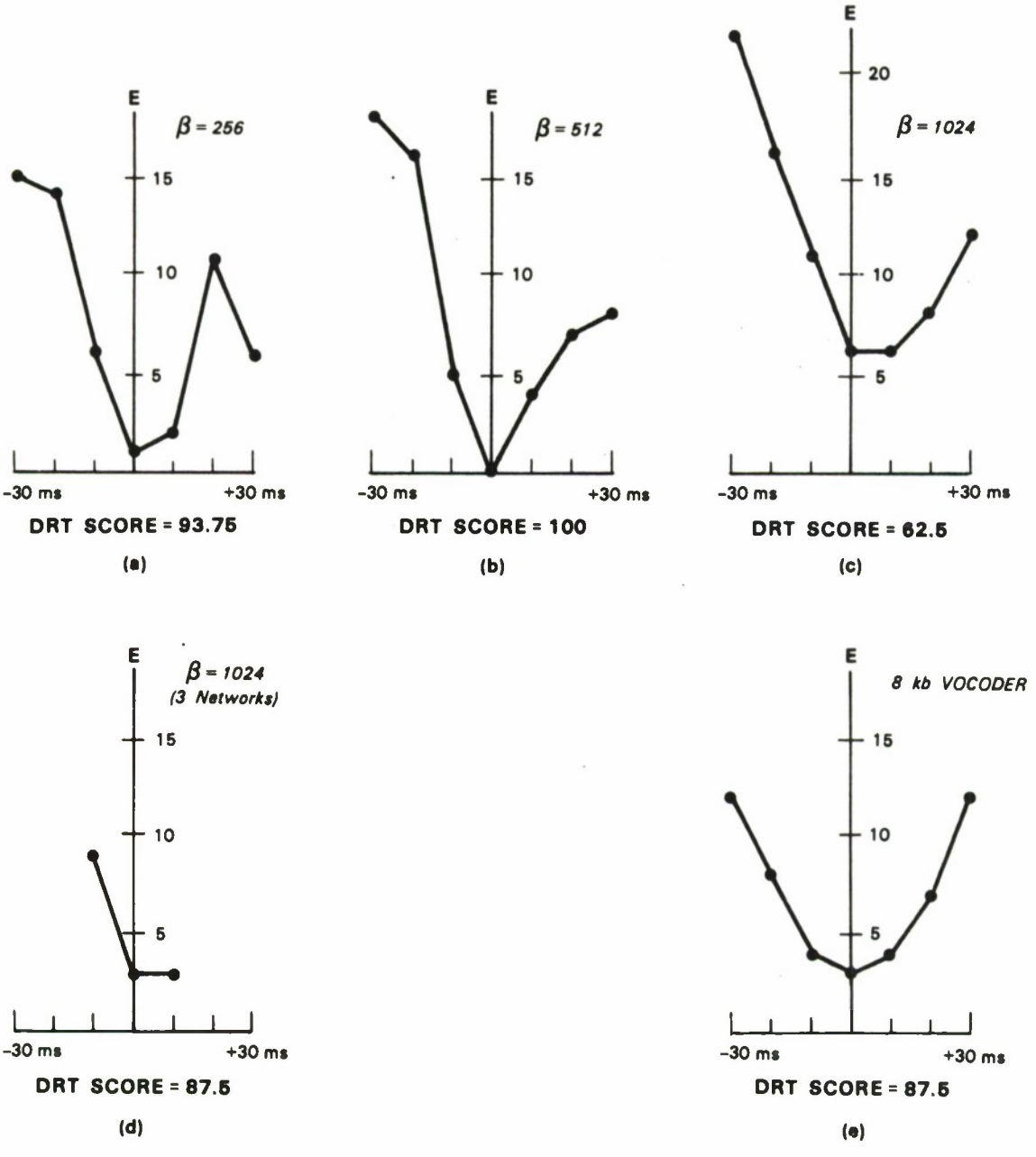
news — dues
fault — vault
coop — poop
sag — shag
teak — peak
keep — cheap
dot — got
boast — ghost
sheet — cheat
pooh — foo
thank — sank
taunt — daunt
thought — fought
moon — noon
tea — key
bend — mend

the critical regions to be used as assigned or stored states. A total of 16 Hopfield nets were thus defined, each with N = 120 and m = 2. Given the word pair from which one of two words would be "heard" by the network allowed us to choose which of these 16 nets to employ.

In Section 6, implementation and simulation issues will be discussed; we wish now to say only that computer running time became rather long so that extensive data were not collected. We did process all 32 words, (a) after passage through a vocoder, and (b) after polluting the speech with additive noise. (Note that the Hopfield nets were derived from the original noiseless data.) Figure 13 shows how well the networks distinguished between the consonants for all 32 words. The curves of Figure 13 were derived by performing a two-state discrimination every 10 ms, using the Hopfield net corresponding to the input word-pair. Zero on the horizontal axis in all cases corresponds to the **critical region**; that segment of the signal used to obtain assigned or stored states. The curves show the total number of errors when all 32 words are processed; Figures 13(a), (b), (c), and (d) are for different degrees of computer-generated white additive noise; Figure 13(e) is for the output of an 8-kbps channel vocoder for noiseless input. We see, in general, increased error rates for decisions made away from the critical region. These (automatic) DRT scores listed near each curve correspond to the results obtained at the critical regions. Figures 13(a), (b), (c), and (e) are all based on the same single network algorithm, but Figure 13(d) is based on a **three**-network algorithm. In addition to a network with stored states from the critical region, there are networks with stored states based on the waveform 20 ms earlier and 20 ms later. Each of the three networks makes a binary decision at its **own** critical region and then a best two out of three determines the final outcome. We see from Figure 13(d) that there was a substantial improvement compared to Figure 13(c), where the additive noise was the same.

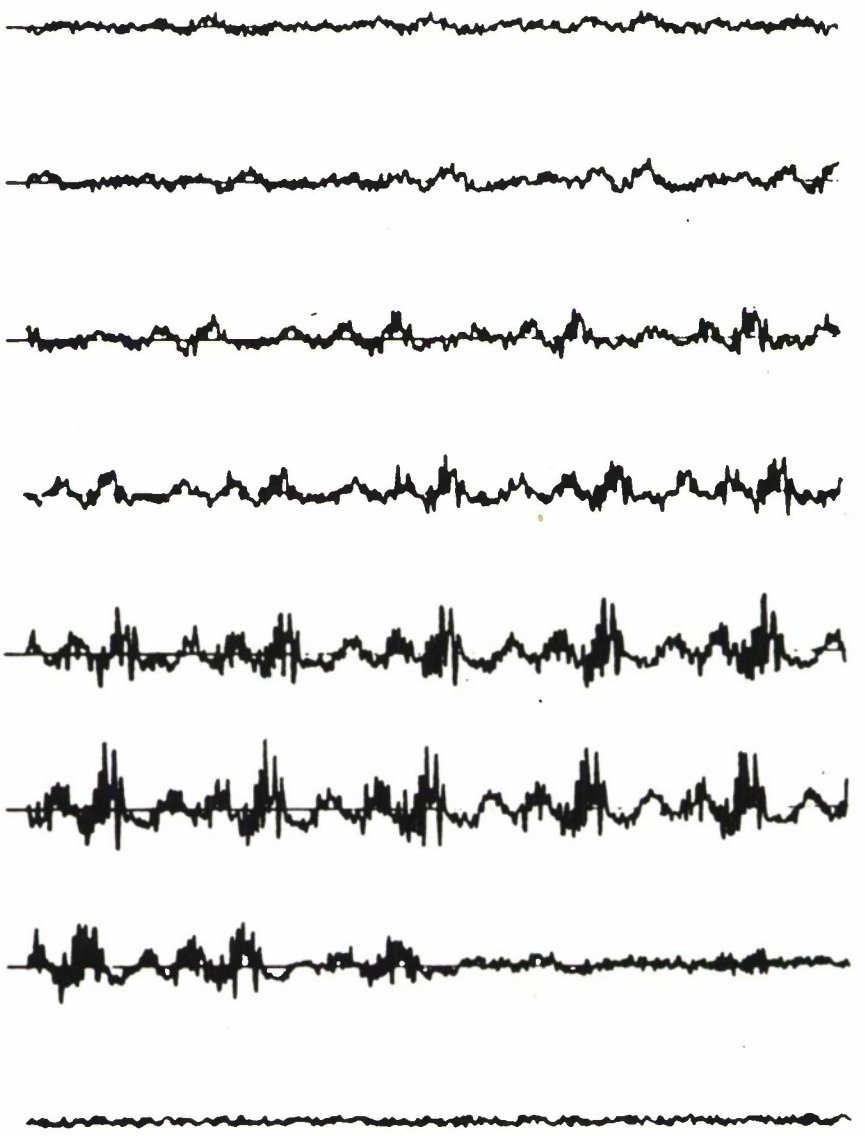
The computer-generated noise was based on an algorithm that generated a uniformly distributed stochastic variable each sample. The parameter β controls the amount of additive noise; in Figure 13, the different values of β that were used are shown for each of the curves. The variance of the noise is $\beta^2/3$; the mean is zero. For the speech plus noise signal of Figure 14, if we make the assumption that the speech is also uniformly distributed in amplitude (and given that the maximum value of the speech is 10140, as shown) we can postulate a 20-dB signal-to-noise ratio during the steady portion of the vowel and perhaps a 0- to 5-dB signal-to-noise ratio during the initial consonant in the word "dues." This ratio corresponds to Figures 13(c) and (d) if we assume an average ratio of vowel to initial consonant of 20 dB.

The same set of 32 DRT words with additive noise with $\beta = 1024$ was scored using three human listeners; the score was 83.3 which is reasonably close to the score obtained using the 3-network configuration, which yielded the result Figure 13(d). This very preliminary result encourages us to continue research on an automatic DRT system using Hopfield nets.



74319-14

Figure 13. Results for Automatic Diagnostic Rhyme Test based on 32 words.



74319-1

Figure 14. Waveform of the word "dues" plus computer-generated noise. Scale is 50 ms/line.

6. SIMULATION OF HOPFIELD NETS

In this section we describe our facility for simulating the algorithm described in the previous sections. We also speculate as to how the facility can be improved.

The LDSP[6] is a 16-bit, fixed-point, general-purpose computer. A typical instruction (including the instruction fetch) is completed in 200 ns, but because of the pipelining feature of the architecture which allows for overlap, a new instruction, most of the time, is available every 50 ns. An exception is the multiply instruction which takes 100 ns. The structure of the simulation program using LDSP is shown in Figure 15. It is important to note that the size of data memory and program memory are 4096 and 2048, respectively. In simulating a Hopfield net of size N, a total of approximately N^2 values of T_{ij} are needed so that for large values of N (100, 200), there is insufficient memory capacity. This problem is to a great extent overcome by a large peripheral memory (LPM). Two of the four available LDSPs have such a peripheral, each with 256K words of storage. As seen in Figure 15, this involves communication between the LDSP and the LPM. This should allow us to simulate networks for N as large as 500 without resorting to the appreciably slower mass memories of the main frame.

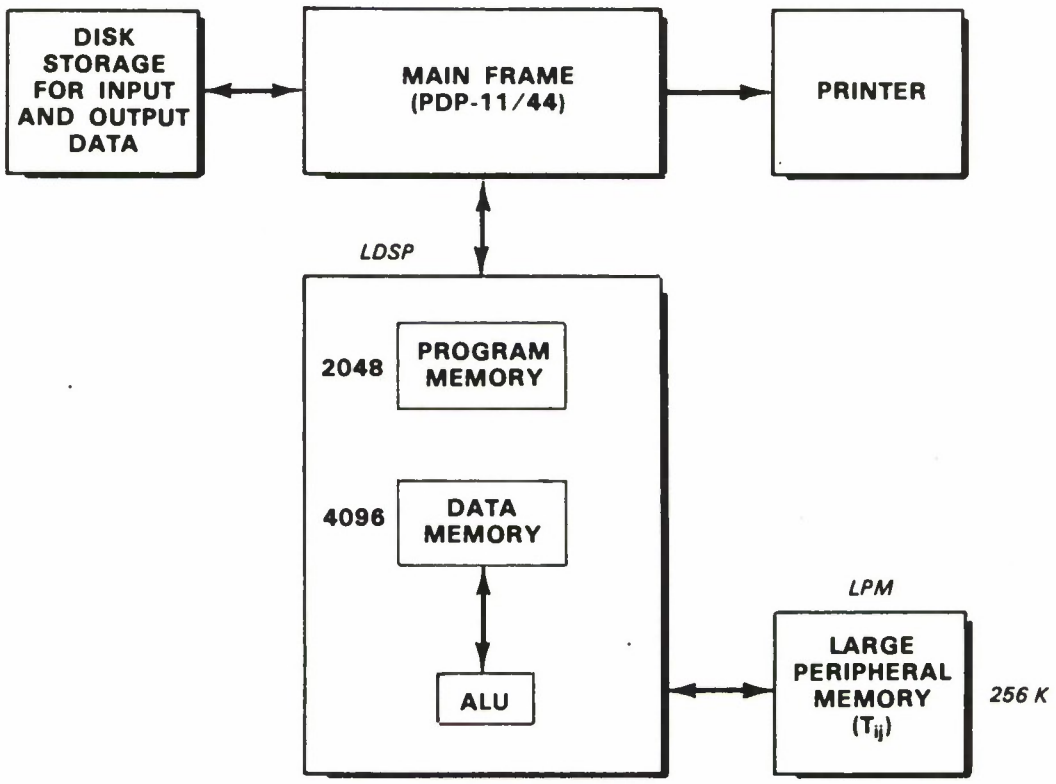


Figure 15. Structure of the Hopfield net simulation using the Lincoln Digital Signal Processors.

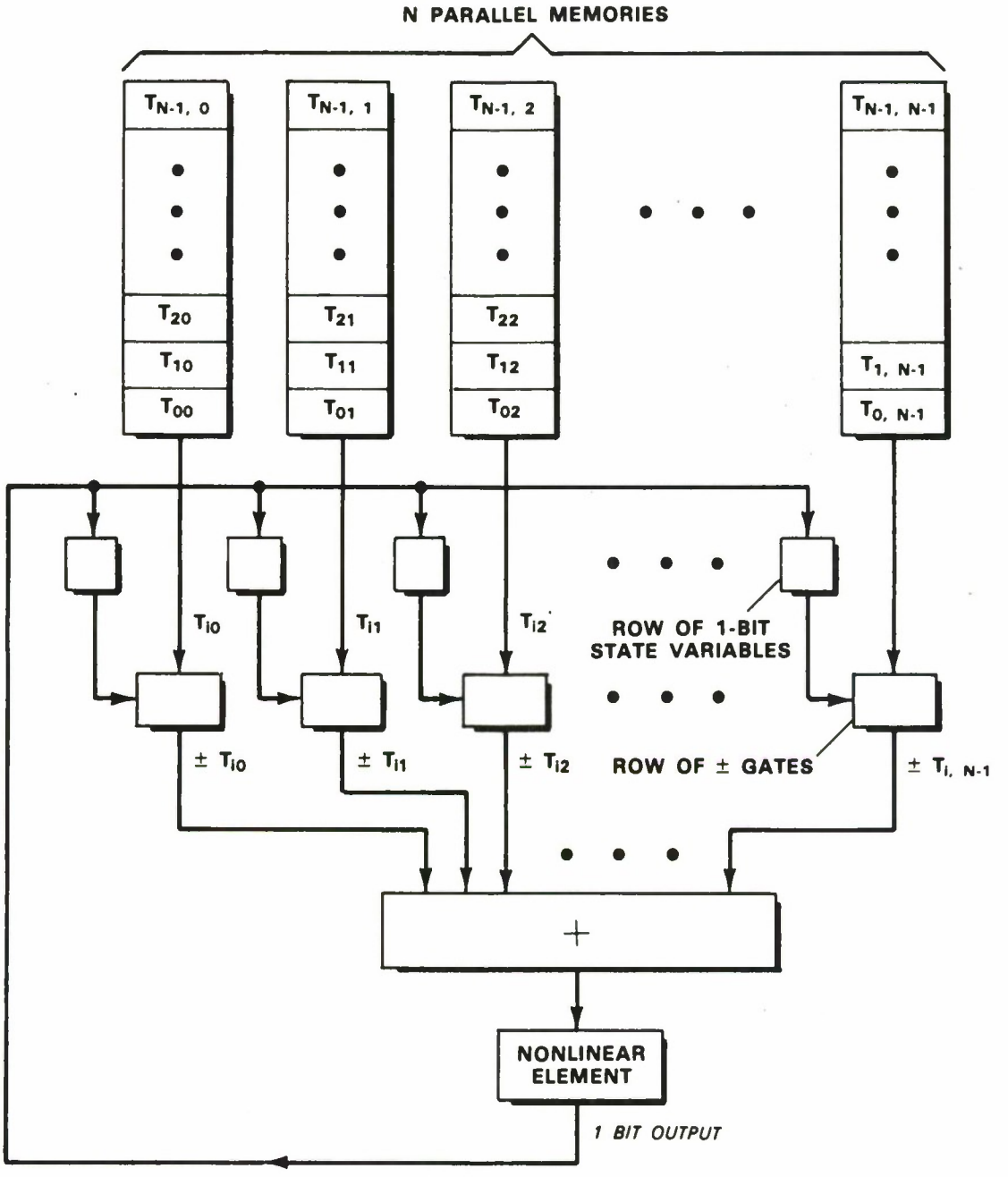


Figure 16. Special peripheral processor to speed up network computations.

Operation of the programs may be divided into two parts: "learning" and "deciding." At the present stage of our knowledge, learning for the problems discussed here is intrinsically a non-real-time operation, involving the judgment of the experimenter in choosing the appropriate states to memorize. For example, in Figure 9, the spectral cross section to be memorized (denoted by the asterisk) is chosen by the experimenter after studying the waveforms and after printing out octal versions of the 120-bit binary representation. Future work should focus on automatic ways for the system to learn the necessary patterns for stored states. Once a set of patterns representing some aspects of either a vowel or consonant is chosen, the T_{ij} matrix must be computed and stored in the LPM, but in our experiments this need be done only **once** for the entire word so that that particular computation [Equation (1)] is an insignificant portion of the entire computing load.

The major computational load is the iteration of the network. Each iteration involves the computation of Equation (2). To compute each new value of $x_i(n)$ takes N operations so that N^2 such operations are needed to compute all the x_i for each iteration. The program is assumed to converge after 32 iterations; experiments have shown that convergence always takes place sooner for the problems we have worked on. With the present program, processing of a 2-s speech utterance [with Equation (2) being applied 32 times every 10 ms] takes about 5 min.; nearly all this time is taken by simulation of the network iteration.

Raffel[7] has suggested a method of increasing the processing speed by the inclusion of a special-purpose processor connected as a peripheral to the LDSP. Figure 16 shows the overall structure of such a proposed device. First, the N memories must be loaded from the LDSP with the T_{ij} matrix. Any particular $x_i(n)$ can be computed in parallel by simultaneously accessing the j-required values of T_{ij}, performing the indicated gating, summation and nonlinear operation, and sending the result back to the i^{th} 1-bit state variable. Assuming that the computation of a single x_i can be carried out in 500 ns, a complete iteration for N = 120 would take 60 μs and 32 iterations would take 1.92 ms. Thus a complete Hopfield net computation performed every 10 ms would run appreciably faster than real time. Also, this structure would allow for the successive computation of several, or many, nets by simply **stacking** T_{ij} matrices in the N parallel memories.

7. DISCUSSION

In the preliminary experiments described in this report, Hopfield networks have proved to be fairly robust implementations of associative memory modules. For speech-processing applications, an appropriate binary representation of the speech is central for obtaining good performance. For different applications (e.g., recognition, vocoding, pitch detection) the representation is bound to be different.

Our attempt to devise an automatic DRT with Hopfield networks indicates that incorporating null states could significantly enhance performance. In a rhyme test such as a DRT, the critical interval of the speech is at or near the consonant-vowel boundary. If the noncritical intervals could be made to converge to such a null state while the critical regions converged to a true memory state, much of the human editing could be eliminated. Several proposals have been made (but not yet tested) on this subject: (a) introduce new "phony" states orthogonal to the "real" states and to each other; convergence to any "phony" state would be interpreted as a null state; (b) introduce a bias in the threshold computation; this would tend to favor, for example, the state component $x_i = -1$ over $x_i = +1$, and the null state could be defined as the state where all the x_i's are -1.

The experiments described in this report are only a beginning toward finding useful applications of neural-network-like structures. Experimental and theoretical studies of such networks have high priority in future efforts in this new and interesting area.

REFERENCES

1. *The Neuro Sciences Fourth Study Program*, F.O. Schmitt and F.G. Worden, Eds. (M.I.T. Press, Cambridge, 1977).
2. T.E. Posch, "Models of the Generation and Processing of Signals by Nerve Cells: A Categorically Indexed Abridged Bibliography," University of Southern California, Electrical Engineering Department Report 290 (August 1968).
3. J.J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Nat. Acad. Sci. USA **79**, 2554-2558 (April 1982).
4. J.J. Hopfield, "Collective Processing and Neural States," CALTECH Chemistry Department, Contribution No. 6802.
5. A.Gelperin, J.J. Hopfield, and D.W. Tank, "The Logic of Limax Learning," in *Model Neural Networks and Behavior*, Edited by A.I. Selverston (Plenum Press, New York, 1985).
6. P.E. Blankenship and V.J. Sferrino, "Succinct LDSP User's Guide," not generally available.
7. J. Raffel, private communication.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-86-018	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Hopfield Model Applied to Vowel and Consonant Discrimination		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER Technical Report 747
7. AUTHOR*(s)* Bernard Gold		8. CONTRACT OR GRANT NUMBER*(s)* F19628-85-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173-0073		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element Nos. 33401F and 64754F
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Systems Command, USAF Andrews AFB Washington, DC 20334		12. REPORT DATE 3 June 1986 13. NUMBER OF PAGES 42
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* Electronic Systems Division Hanscom AFB, MA 01731		15. SECURITY CLASS. *(of this Report)* Unclassified 15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT *(of this Report)* Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*		
18. SUPPLEMENTARY NOTES None		
19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)* neural networks Hopfield networks associative memory vowel discrimination consonant discrimination Automatic Diagnostic Rhyme Test		

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The model was able to converge to the correct vowel (out of 8) 51 out of 64 times even though only half of the pattern was specified. Also, the model was able to recognize "familiar" vowels from the same utterance. Finally, the model was configured to perform a simplified Automatic Diagnostic Rhyme Test.